Addendum to Supplementary Materials, Part 2, for the Blueprint of the "Alignment Neighborhood Explorer" (ANEX) (tentatively named),

by Kiyoshi Ezawa

(Finished on January 25th, 2018; CC4 statement added on August 14th, 2020)

Table of Contents

Supplementary-Supplementary Results & Discussion

SSR-1. Revisiting the characterization of "purge"-like errors

In section SR-1 of "sppl2_blueprint1_ANEX.xxxx.doc", we characterized the "purge"-like errors in terms of the P-value of the substitutional difference (under a given base substitution model), the size of blocks defining the "purge" (or the control), etc. (See tables SS3-SS5 of "sppl2_blueprint1_ANEX.xxxx.doc".)

These results indicated that the P-value will quite efficiently identify the candidate regions that are likely to contain the "purge"-like errors. Thus, I created some Perl modules, e.g., an *old version* of "detect_purge_cands" in "MyDetect_purge_cands.pm", to detect such candidate regions via a sliding-window analysis of the aforementioned P-value.

When applying the modules to some sample MSAs, however, I noticed that the set of candidate windows consists mostly of "false positives", each of which contains only one substitutional difference (especially along a relatively short branch).

It would therefore be desirable if we have a method to filter a substantial fraction of such "false positives" while keeping most of true positives.

One such way would be to filter the windows via **another P-value** (again regarding the substitutional difference), which is defined under a random-matching model.

The rationale for this is as follows. First, the "purge" errors are expected to occur in general by falsely eliminating a pair of neighboring complementary indels at the expense of generating false substitutions. Thus, roughly speaking, the "false-homologous blocks" caused by a "purge" should be like an alignment of two random segments (or two non-homologous sub-alignments). The original P-value attempts to identify "likely false-homologous pairs" that show significantly more substitutional differnces than expected under a given substitution model. Along a short branch, however, this measure is likely to pick even a window showing only one (or two) substitution(s). In contrast, the new P-value attempts to identify "likely true-homologous pairs" that show significantly less substitutional differences than expected under the random matching model, like BLAST. Because the random-matching model is nearly independent of the branch length, the new P-value is expected to give an effectively "orthogonal" filtering to that via the original P-value, even though both of them are defined in terms of the same measure, i.e., the substitutional difference.

I incorporated the filtering via this new P-value into the new version of **"detect_purge_cands"** in **"MyDetect_purge_cands.pm"**.

It would be prudent, however, to examine whether our expectation is indeed true or not.

Thus, I examined the 2-dimensional distributions of the old P-value vs the new P-value, one

calculated on the true "purge"-errors (subjects) and the other on the correctly aligned regions (controls).

Tables SSS1-SSS3(?) summarize the results.

First, as Table SSS1 shows, discarding those windows with significantly more matches than random does indeed refine the screening via a base substitution model alone, by keeping most of true-positives (in the 1st screening) and by shedding a substantial fraction, from a near majority to an overwhelming majority, of false-positives (again in the 1st screening).

Tables SSS2 and SSS3 indicate that such an additional screening becomes more effective if we exclude those windows with size 1 or 2 from the window analysis.

If we include such "small" windows into the window analysis, we are likely to end up considering the potential "purge"-errors of nearly all those sites showing substitutional differences, which could be very time-consuming. Therefore, until a very fast algorithm is invented to examine potential "purge"s, excluding "small" windows from the analysis would considerably save time. If such a practice is employed, the additional screening based on the random-matching model would be more beneficial.

Supplementary-Supplementary Tables SSS1-SSS3

Table SSS1. Effects of additional filtering via the new P-value (on all subjects & controls).

	Subjects		**Controls**	
	P(sbst) < 0.05	**P(sbst) < 0.20**	**P(sbst) < 0.05**	**P(sbst) < 0.20**
(No further condition)	0.495	0.780	0.0088	0.057
P(rand) ≥ 0.05	0.478	0.747	0.0061	0.031
P(rand) ≥ 0.20	0.433	0.664	0.0042	0.018

NOTE: Shown in each cell is the relative frequency of "purge"-involved blocks (in the subjects) or windows (in the controls) satisfying the specified condiiton, in the set of reconstructed MSAs of 15 simulated mammalian sequences. The "P(sbst)" and "P(rand)" stand for, respectively, the (old) P-value defined with a given base substitution model and the (new) P-value defined with the random matching model.

Table SSS2. Effects of additional filtering via the new P-value (on subjects & controls with block size ≥ 2).

	Subjects		**Controls**	
	P(sbst) < 0.05	**P(sbst) < 0.20**	**P(sbst) < 0.05**	**P(sbst) < 0.20**
(No further condition)	0.590	0.812	0.0092	0.060
P(rand) ≥ 0.05	0.569	0.773	0.0059	0.028
P(rand) ≥ 0.20	0.515	0.673	0.0035	0.013

NOTE: The same note applies as that for Table SSS1. The only difference with Table SSS1 is that the statistics here exclude subjects and controls with block size 1.

Table SSS3. Effects of additional filtering via the new P-value (on subjects & controls with block size ≥ 3).

	Subjects		**Controls**	
	P(sbst) < 0.05	**P(sbst) < 0.20**	**P(sbst) < 0.05**	**P(sbst) < 0.20**
(No further condition)	0.649	0.895	0.0094	0.065
P(rand) ≥ 0.05	0.623	0.844	0.0054	0.027
P(rand) ≥ 0.20	0.552	0.713	0.0025	0.0076

NOTE: The same note applies as that for Table SSS1 (& Table SSS2). The only difference with Table SSS1 (& Table SSS2) is that the statistics here concerns only those subjects and controls with block size 3 or greater.